Given the `Vector` class implementation from the Princeton Algorithms Library, we can now clearly describe the behavior of the method calls `x.add(y)` and `y.add(x)` when `x` corresponds to the vector $(1, 2, 3)$ and `y` corresponds to the vector $(5, 6)$.

Behavior of `x.plus(y)` and `y.plus(x)`

The `Vector` class has a `plus` method which is used to add two vectors. Here's the method's implementation:

```
// return this + that
public Vector plus(Vector that) {
    if (this.length() != that.length())
        throw new IllegalArgumentException("dimensions disagree");
    Vector c = new Vector(n);
    for (int i = 0; i < n; i++)
        c.data[i] = this.data[i] + that.data[i];
    return c;
}
```

This method checks if the lengths of the two vectors are the same and throws an `IllegalArgumentException` if they are not. If the lengths are the same, it creates a new `Vector` instance where each element is the sum of the corresponding elements from `this` vector and `that` vector.

Given:

- $x = (1, 2, 3)$
- $y = (5, 6)$

When we call `x.plus(y)`:

1. The `plus` method checks if the lengths of `x` and `y` are the same.
2. `x.length()` is 3 and `y.length()` is 2.
3. Since the lengths are not the same, an `IllegalArgumentException` will be thrown with the message "dimensions disagree".

When we call `y.plus(x)`:

1. The `plus` method checks if the lengths of `y` and `x` are the same.
2. `y.length()` is 2 and `x.length()` is 3.
3. Since the lengths are not the same, an `IllegalArgumentException` will be thrown with the message "dimensions disagree".

Conclusion

For both method calls `x.plus(y)` and `y.plus(x)` with the given vectors $x = (1, 2, 3)$ and $y = (5, 6)$:

- The calls will result in an `IllegalArgumentException` because the vectors have different lengths.

Here is a summary of what happens for each call:

- **`x.plus(y)`**: Throws `IllegalArgumentException` due to length mismatch (3 vs 2).
- **`y.plus(x)`**: Throws `IllegalArgumentException` due to length mismatch (2 vs 3).

This behavior ensures that vector addition only occurs between vectors of the same dimension, which is a fundamental requirement for vector operations in linear algebra.